

# Dynamische Arrays (A)

Bei dieser Station lernst du die Listen-Struktur und die zugehörigen Manipulationsmethoden, die der MIT App Inventor Listen bietet, kennen und wendest sie in einer simplen Einkaufs- oder Todolisten-App an.

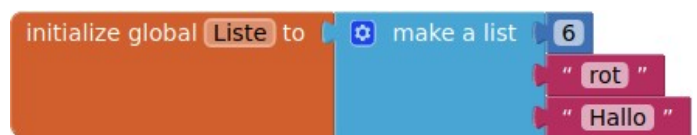
**Theorie:** Der App Inventor bietet die Datenstruktur *List* an, die einem dynamischen Array (auch Array-Liste genannt) entspricht. D.h. eine solche Liste speichert eine variable Anzahl an Werten, auf die man, beginnend mit 1, per Index zugreifen kann. Es gibt Programmierblöcke, mit denen man die Liste am Ende oder an einer bestimmten Position erweitern kann, Elemente entfernen oder ersetzen kann, aber auch komplexere vorgefertigte Programmierbefehle z.B. zum Import einer csv-Tabelle. All diese Befehle findest du in der Kategorie *Lists*.

Nachstehend werden die Begriffe *Array*, *dynamisches Array* und *Liste* synonym für die erwähnte Struktur verwendet.

## Aufgabe 1 – Liste erstellen, Element einfügen und entfernen

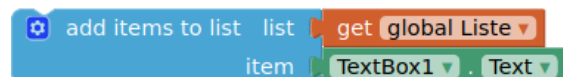
Erstelle ein Projekt mit zwei Textfeldern, einem Label und zwei Buttons.

- a) Initialisiere eine globale Variable als Liste mit den Werten rechts. Den *make a list*-Block kann man mit dem Einstellungs-rädchen erweitern. Der Name der Variable ist in der Grafik oben *Liste*, könnte aber natürlich auch *Gabriele* sein.

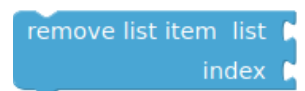


Beim Klick auf den Button soll der Wert der Variable (also die gesamte Liste) als Text des Labels angezeigt werden: `[6, "rot", "Hallo"]`

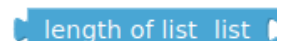
- b) Derselbe Button soll dafür sorgen, dass der Text aus dem ersten Textfeld als Wert ans Ende der Liste angefügt wird und die Liste dann wieder als Text des Labels ausgegeben wird.



- c) Der zweite Button soll dafür sorgen, dass das x-te Element aus der Liste entfernt wird, wobei man den Wert x im zweiten Textfeld einträgt. Wenn dort also eine 2 steht, soll das zweite Element der Liste entfernt werden. Anschließend soll die Liste zur Überprüfung als Text des Labels ausgegeben werden.



- d) **Bonus:** Die Methode, die vom Button aufgerufen wird, überprüft selbst, ob der Index in der Liste existiert, also nicht größer als die Länge der Liste ist.



- e) **Bonus:** Es wird vorher überprüft, ob der Wert eine natürliche Zahl ist.

**Aufgabe 2 – Einkaufs- oder Todoliste**

Erstelle ein Projekt mit einem Textfeld, einem Button und einem Listview-Objekt. Trage über die Properties des Listview-Objekts drei Elemente ein.



Wasser, Oliven, Kekse

Hierdurch wird automatisch eine globale Variable `Listview1.Elements` (wenn das Listview-Objekt `Listview1` heißt) deklariert und mit den Einträgen initialisiert.

- a) Der Button sorgt dafür, dass das, was im Textfeld steht, der Liste hinzugefügt wird.

Tipp: Es reicht, der Elemente-Liste ein neues Element anzufügen und mit folgendem Befehl die neue Liste anzuzeigen:

```
set ListView1 . Elements to ListView1 . Elements
```

- b) Wenn ein Element des Listview-Objekts angeklickt wird (*AfterPicking*), soll es entfernt werden. Tipp: Der Index des gewählten Elements steht im *SelectionIndex* des Objekts.

# Dynamische Arrays (B)

Diese Station baut auf *Dynamische Arrays (A)* auf und dreht sich darum, dynamische Arrays mit Hilfe einer Schleife zu befüllen, den größten oder durchschnittlichen Wert zu bestimmen und Listen aus einem String zu erstellen.

## Aufgabe 1 – Liste befüllen

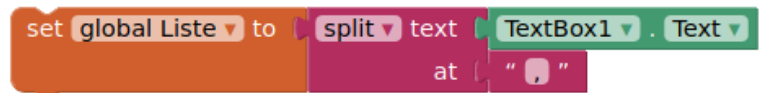
Für die nachfolgenden Aufgaben ist es praktisch, schnell mit Zahlen gefüllte Arrays erstellen zu können. Hier lernst du zwei Optionen kennen.

Erstelle ein Projekt mit einem Textfeld, einem Label und einem Button. Initialisiere außerdem eine neue Liste als leere Liste.

- Beim Klick auf den Button soll das Array mit den Zahlen von 1 bis X befüllt werden, wobei X die Zahl in dem Textfeld ist.

Tipp: Verwende eine for-Schleife (*for each number from...to...by...*).

- Beim Klick auf den Button soll der Text im Textfeld als Komma-getrennte Liste eingelesen werden. D.h. wenn dort 2,45,3 steht, wird eine Liste ["2","45","3"] generiert und als Labeltext ausgegeben.

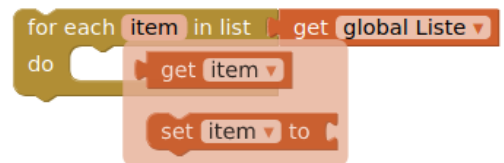


## Aufgabe 2 – Schleife über alle Listenelemente

Verwende dein Projekt aus Aufgabe 1a oder b.

- Beim Klick auf den Button wird der größte Wert einer generierten Liste (egal nach welcher Methode) angezeigt. Gehe hierbei davon aus, dass alle Werte der Liste numerisch sind.

Tipp: Es gibt eine Schleife, mit der man über alle Elemente (items) einer Liste iterieren kann. Wenn man mit der Maus über *item* fährt, wird einem der zugehörige get- bzw. set-Befehl angezeigt, der in jedem Schleifendurchlauf auf ein neues Listenelement zugreift.



Versuche, für das größte Element eine lokale Variable anstatt einer globalen zu verwenden.

- Füge einen weiteren Button ein, der dafür sorgt, dass alle ungeraden Werte in der Liste verdoppelt werden.
- Füge einen weiteren Button ein, mit dem der Durchschnitt der Listeneinträge bestimmt wird.
- Bonus:** Falls du dich mit Sortieralgorithmen auskennst, füge einen Button ein, der die Liste alphabetisch sortiert. Tipp: Unter den Text-Befehlen gibt es einen alphabetischen Vergleichsoperator.